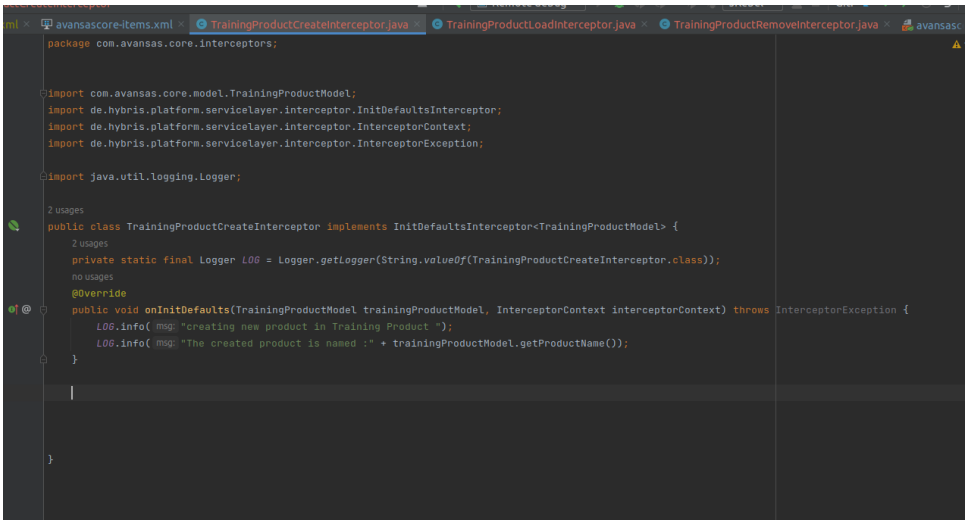
Ahmat Barkai Moussa

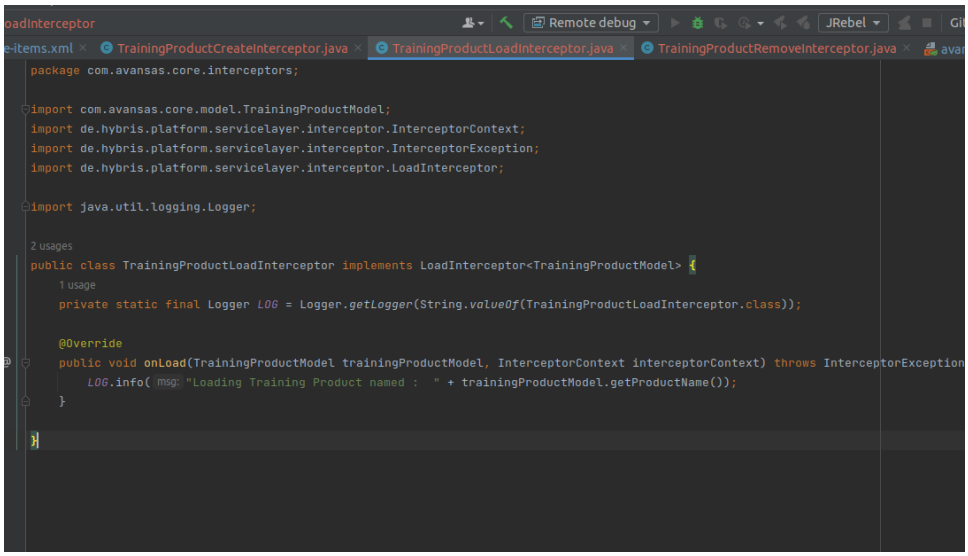
Task 8 : Interceptor Hazırlanması : Oluşturduğunuz tablo üzerinde CRUD operasyonları gerçekleştiği an araya giren ve terminale log atan interceptor'ları yazın. 5 interceptor'ı da araştırın ve kısaca ne yaptıklarından bahsedin.

1 – create için :interceptor package'inde TrainingProductCreateInterceptor oluşturuyoruz ve InitDefaultsInterceptor'ı implements ederek modelimizi parametre olarak veriyoruz . OnInitDefaults metodu override edip içinde log ile terminale mesaj veriyoruz yeni bir product create ettiğimiz an .

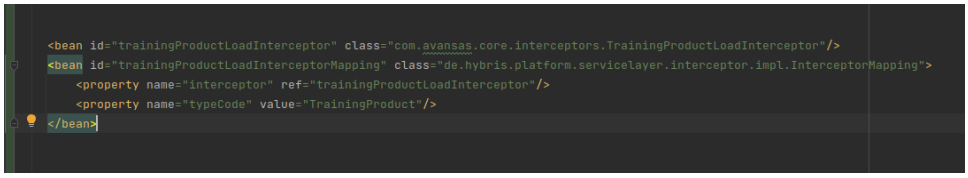


Ve avansascore-spring.xml'de yeni oluşturduğumuz class'ı tanımlıyoruz ve interceptorumuza da dahil ediyoruz ,interceptor mapping'de ve typecode'u ise modelimizi veriyoruz (tableName).

2-Veri tabanından veri çekildiği an LoadInterceptor çalışması için interceptor package'inde TrainingProductLoadInterceptor sınıfı oluşturuyoruz ve `LoadInterceptor'ı` implements ederek modelimizi parametre olarak veriyoruz . Onload metodu override edip içinde log ile terminale mesaj verilir bir product veritanından çekildiği an .



Ve avansascore-spring.xml'de yeni oluşturduğumuz class'ı tanımlıyoruz ve interceptorumuza da dahil ediyoruz ,interceptor mapping'de ve typecode'u ise modelimizi veriyoruz (tableName).



3-Product silinmek intendiği an remove interceptoru oluşturuyoruz : TrainingProductRemoveInterceptor sınıf oluşturuyoruz ve RemoveInterceptor interface'ı implements ediyoruz ve parametre olarak TrainingProductModel'i veriyoruz onRemove motodu override edip içinde bussiness logic'lerimizi yazıyoruz .

```java
package com.avansas.core.interceptors;

import com.avansas.core.model.TrainingProductModel;
import de.hybris.platform.servicelayer.interceptor.InterceptorContext;
import de.hybris.platform.servicelayer.interceptor.InterceptorException;
import de.hybris.platform.servicelayer.interceptor.RemoveInterceptor;

import java.util.logging.Logger;

2 usages
public class TrainingProductRemoveInterceptor implements RemoveInterceptor<TrainingProductModel> {
    1 usage
    private static final Logger LOG = Logger.getLogger(String.valueOf(TrainingProductRemoveInterceptor.class));

    @Override
    public void onRemove(TrainingProductModel trainingProductModel, InterceptorContext interceptorContext) throws InterceptorException {
        LOG.info( msg: "Removing Training Product named :  " + trainingProductModel.getProductName());
    }

}
```

Oluşturduğumuz class için ve interceptor için gereken beans'leri yazıyoruz.(interceptorMapping kullanarak)

```xml
    <bean id="trainingProductRemoveInterceptor" class="com.avansas.core.interceptors.TrainingProductRemoveInterceptor"/>
    <bean id="trainingProductRemoveInterceptorMapping" class="de.hybris.platform.servicelayer.interceptor.impl.InterceptorMapping">
        <property name="interceptor" ref="trainingProductRemoveInterceptor"/>
        <property name="typeCode" value="TrainingProduct"/>
    </bean>
```

4-Update ve Validate için :

```java
package com.avansas.core.interceptors;

import com.avansas.core.model.TrainingProductModel;
import de.hybris.platform.servicelayer.interceptor.InterceptorContext;
import de.hybris.platform.servicelayer.interceptor.InterceptorException;
import de.hybris.platform.servicelayer.interceptor.PrepareInterceptor;

import java.util.logging.Logger;

2 usages
public class TrainingProductUpdateInterceptor implements PrepareInterceptor<TrainingProductModel> {
    1 usage
    private static final Logger LOG = Logger.getLogger(String.valueOf(TrainingProductUpdateInterceptor.class));

    @Override
    public void onPrepare(TrainingProductModel trainingProductModel, InterceptorContext interceptorContext) throws InterceptorException {
        LOG.info( msg: "Updating Training Product named :  " + trainingProduct
    }
}
```

Fetching Documentation...

```java
package com.avansas.core.interceptors;

import com.avansas.core.model.TrainingProductModel;
import de.hybris.platform.servicelayer.interceptor.InterceptorContext;
import de.hybris.platform.servicelayer.interceptor.InterceptorException;
import de.hybris.platform.servicelayer.interceptor.ValidateInterceptor;

import java.util.logging.Logger;

2 usages
public class TrainingProductValidateInterceptor implements ValidateInterceptor<TrainingProductModel> {

    1 usage
    private static final Logger LOG = Logger.getLogger(String.valueOf(TrainingProductValidateInterceptor.class));

    @Override
    public void onValidate(TrainingProductModel trainingProductModel, InterceptorContext interceptorContext) throws InterceptorException {
        LOG.info( msg: "Validating Training Product named :  " + trainingProductModel.getProductName());
    }
}
```

Beans'leri tanımlıyoruz

```xml
<bean id="trainingProductUpdateInterceptor" class="com.avansas.core.interceptors.TrainingProductUpdateInterceptor"/>
<bean id="trainingProductUpdateInterceptorMapping" class="de.hybris.platform.servicelayer.interceptor.impl.InterceptorMapping">
    <property name="interceptor" ref="trainingProductUpdateInterceptor"/>
    <property name="typeCode" value="TrainingProduct"/>
</bean>


<bean id="trainingProductValidateInterceptor" class="com.avansas.core.interceptors.TrainingProductValidateInterceptor"/>
<bean id="trainingProductValidateInterceptorMapping" class="de.hybris.platform.servicelayer.interceptor.impl.InterceptorMapping">
    <property name="interceptor" ref="trainingProductValidateInterceptor"/>
    <property name="typeCode" value="TrainingProduct"/>
</bean>

<!--Interceptor crud beans finished -->
```

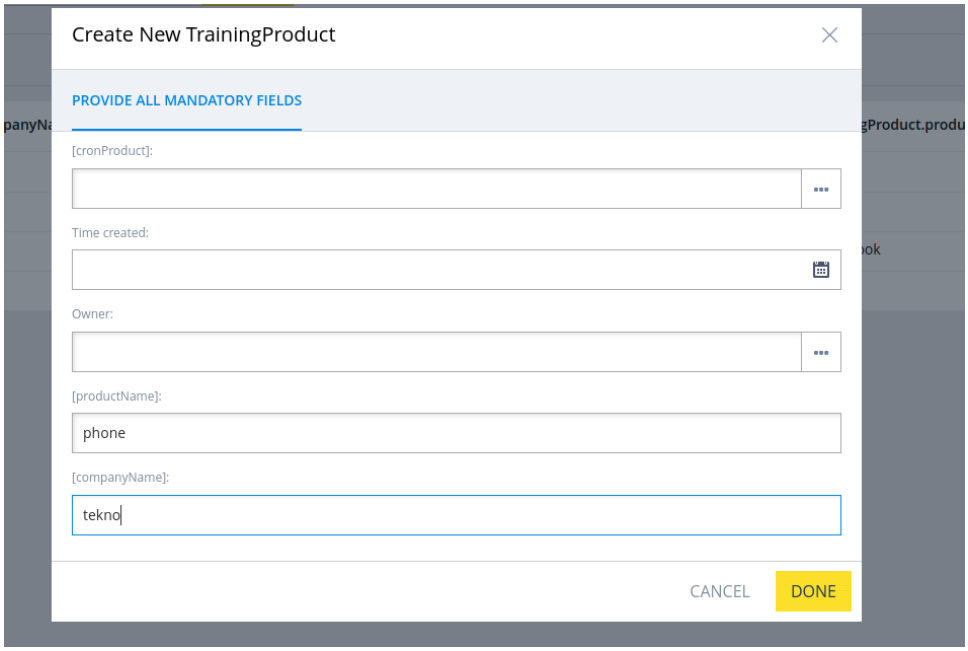
Ant clean ve serveri ayağa kalırıyoruz .

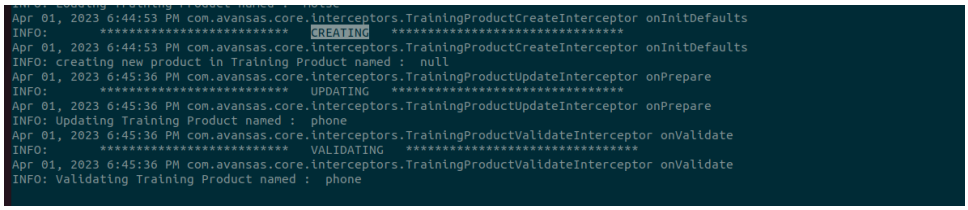
<u>Sonuçlar</u> :

Database'e tablomuza ulaşınca terminale mesajımızı gelmektedir :

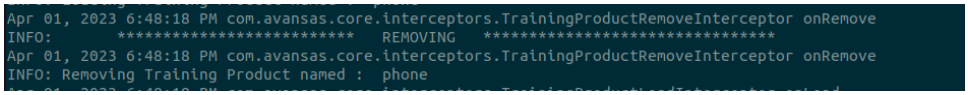
TrainingProduct.companyName	PK	TrainingProduct.productName
amour	8796322459652	oil
apple	8796224155652	mackbook
samsun	8796191387652	moise

```
INFO  [hybrisHTTP2] [LabelLoaderImpl] Opening jar:file:/home/barkai/enoca/avansasproject/bin/ext-backoffice/backof

INFO  [hybrisHTTP6] [LabelLoaderImpl] Loading labels for en_US
INFO  [hybrisHTTP6] [LabelLoaderImpl] Loading labels for en
INFO  [hybrisHTTP6] [LabelLoaderImpl] Opening jar:file:/home/barkai/enoca/avansasproject/bin/ext-backoffice/backof

WARN  [hybrisHTTP18] [ListViewCollectionBrowserMoldStrategy] Missing Column Configuration for type:

Apr 01, 2023 6:41:17 PM com.avansas.core.interceptors.TrainingProductLoadInterceptor onLoad
INFO:       *************************   LOADING   ******************************
Apr 01, 2023 6:41:17 PM com.avansas.core.interceptors.TrainingProductLoadInterceptor onLoad
INFO: Loading Training Product named :  oil
Apr 01, 2023 6:41:17 PM com.avansas.core.interceptors.TrainingProductLoadInterceptor onLoad
INFO:       *************************   LOADING   ******************************
Apr 01, 2023 6:41:17 PM com.avansas.core.interceptors.TrainingProductLoadInterceptor onLoad
INFO: Loading Training Product named :  mackbook
Apr 01, 2023 6:41:17 PM com.avansas.core.interceptors.TrainingProductLoadInterceptor onLoad
INFO:       *************************   LOADING   ******************************
Apr 01, 2023 6:41:17 PM com.avansas.core.interceptors.TrainingProductLoadInterceptor onLoad
INFO: Loading Training Product named :  moise
```





Product Sildiğimiz an :



Debuglarayak tek tek fonksiyonlarımıza düşüyor .

Görsellerden sonra
LOG.info(" ********** CREATING ********** ");

Fonksiyonu eklenmiş.

II- 5 Interceptor'ların açıklaması :

1-InitDefaultInterceptor : bu interceptor yeni bir model oluşturulduğu an çağırılmaktadır , modelService.create(model) kullanarak ve onInitDefault metodunu override ediyor .

2-LoadInterceptor : database'ten veri çekilince çağırılmaktadır .

modelService.get()'i kullanıyor. Bu interceptor onLoad() metodu override ediyor .

3-RemoveInterceptor : Bu interceptor database’te veri silinice çalışmaktadır (modelService.remove()) bu ise onRemove metodunu override ediyor .

4-PrepareInterceptor : veri update edince çalışmaktadır ve onPrepare metodunu override ediyor .

5-Ve ValidateInterceptor : Hemen prepareInterceptor’dan sonra yer almaktadır veri tabanına save etmek istediğimizi kontrol edilip öyle veri tanına kayıt edililmektedir .Bu interceptor onValidate kullanmaktadır .